WHAT IS CLAIMED IS:

1. A method of validating and dispatching an event, comprising:

receiving an event;

determining an exception handler for the event;

determining if the exception handler is valid; and

executing the exception handler if the exception handler is valid.

2. The method of claim 1, wherein determining if the exception handler is valid comprises comparing the exception handler to a list of valid exception handlers.

3. The method of claim 2, further comprising one of receiving and generating the list of valid exception handlers.

4. The method of claim 1, further comprising retrieving a list of valid exception handlers from a storage device and comparing the exception handler to the list of valid exception handlers in determining if the exception handler is valid.

5. The method of claim 1, further comprising generating a list of valid exception handlers by compiling code into at least one of an object file and an image.

6. The method of claim 1, further comprising compiling code to produce an executable that is marked with an identifier indicating that the executable is safe with respect to a list of valid exception handlers.

7. The method of claim 1, further comprising, if the exception handler is valid, determining whether the exception handler handles the event, and if so, executing the exception handler, and otherwise, retrieving a second exception handler from information on a stack and continuing processing with determining if the second exception handler is valid.

8. The method of claim 1, further comprising terminating the method if the handler is invalid.

9. The method of claim 1, further comprising generating an error message if the handler is invalid.

10. The method of claim 1, further comprising, if the exception handler is valid, verifying other data for the event.

11. The method of claim 10, wherein the other data comprises pointer data.

12. A computer-readable medium having stored thereon computer-executable instructions for performing a method of validating and dispatching an event, the method comprising:

receiving an event;

determining an exception handler for the event;

determining if the exception handler is valid; and

executing the exception handler if the exception handler is valid.

13. The computer-readable medium of claim 12, wherein determining if the exception handler is valid comprises comparing the exception handler to a list of valid exception handlers.

14. The computer-readable medium of claim 13, having further computer-executable instructions for one of receiving and generating the list of valid exception handlers.

15. The computer-readable medium of claim 12, having further computer-executable instructions for retrieving a list of valid exception handlers from a storage device and comparing the exception handler to the list of valid exception handlers in determining if the exception handler is valid.

16. The computer-readable medium of claim 12, having further computer-executable instructions for generating a list of valid exception handlers by compiling code into at least one of an object file and an image.

17. The computer-readable medium of claim 12, having further computer-executable instructions for compiling code to produce an executable that is marked with an identifier indicating that the executable is safe with respect to a list of valid exception handlers.

18. The computer-readable medium of claim 12, having further computer-executable instructions for, if the exception handler is valid, determining whether the exception handler handles the event, and if so, executing the exception handler, and otherwise, retrieving a second exception handler from information on a stack and continuing processing with determining if the second exception handler is valid.

19. The computer-readable medium of claim 12, having further computer-executable instructions for terminating the method if the handler is invalid.

20. The computer-readable medium of claim 12, having further computer-executable instructions for generating an error message if the handler is invalid.

21. The computer-readable medium of claim 12, having further computer-executable instructions for, if the exception handler is valid, verifying other data for the event.

22. The computer-readable medium of claim 21, wherein the other data comprises pointer data.

23. A system for validating an event to be dispatched, comprising:

a processor that receives an event; and

an exception dispatcher system that determines an exception handler for the event, determines if the exception handler is valid; and executes the exception handler if the exception handler is valid.

24. The system of claim 23, wherein the exception dispatcher system determines if the exception handler is valid by comparing the exception handler to a list of valid exception handlers.

25. The system of claim 23, further comprising a storage device that stores a list of valid exception handlers, and the exception dispatcher system retrieves the list of valid exception handlers from the storage device and compares the exception handler to the list of valid exception handlers in determining if the exception handler is valid.

26. The system of claim 23, further comprising at least one of a compiler and an assembler that generates a list of valid exception handlers in at least one of an object file and an image.

27. The system of claim 26, further comprising a storage device that stores the list of valid exception handlers.

28. The system of claim 23, further comprising a compiler and a linker that compiles code to produce an executable that is marked with an identifier indicating that the executable is safe with respect to a list of valid exception handlers.

29. The system of claim 23, wherein the exception dispatcher system, if the exception handler is valid, determines whether the exception handler handles the event, and if so, executes the exception handler, and otherwise, retrieves a second exception handler from information on a stack and continues processing with determining if the second exception handler is valid.

30. The system of claim 23, wherein the exception dispatcher system terminates processing if the handler is invalid.

31. The system of claim 23, wherein the exception dispatcher system generates an error message if the handler is invalid.

32. The system of claim 23, further comprising a linker that creates an image based on at least one object file received from at least one of a compiler and an assembler, and provides the image to the exception dispatcher system.

33. The system of claim 32, wherein the linker produces an executable that is marked with an identifier indicating that the executable is safe with respect to a list of valid exception handlers.

34. The system of claim 23, wherein the exception dispatcher system, if the exception handler is valid, verifies other data for the event.

35. The system of claim 34, wherein the other data comprises pointer data.